DOCUMENT RESUME

ED 383 168 FL 022 911

AUTHOR Mitschian, Haymo
TITLE Status- und Steuerelemente fur sprachbezoegene Ubungsprogramme (Status and Control Elements for Language Related Training Program).
PUB DATE Sep 94
NOTE 18p.; Paper presented at EUROCALL (1994).
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)
LANGUAGE German

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Software; *German; Languages for Special Purposes; *Professional Training; Second Language Instruction; Standards

ABSTRACT

To use computers as a medium to learn a language, there is need for some GUI-elements indicating and regulating the actual status of training. These elements have to be separated into obligatory ones, useful in almost all CALL-software, and facultative ones, necessary for most "teachware." Due to a general need of reducing the difficulties to get into the use of a training program for language learning, a standardization, as far as possible, of these GUI-elements should be aimed for. Suggestions for design and function are made on the basis of the training program "Verbindungsarten" (German as a foreign language/for special purposes: mechanical engineering). Pictures of actual computer screens are included. (Contains 3 references.) (Author)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 383 168

Haymo Mitschian

Status- und Steuerelemente für sprachbezogene Übungsprogramme

Abstract

Die Eigenschaften des Lernmediums Computer erfordern für das sprachbezogene Üben eine Reihe von Oberflächenelementen zur Statusanzeige und zur Übungssteuerung, wobei zwischen obligatorischen, für (fast) alle (Sprach-)Lernprogramme nötigen und fakultativen, meist oder häufig einsetzbaren Elementen unterschieden werden kann. Mit Blick auf die Notwendigkeit zur Reduzierung des Einarbeitungsaufwandes in Lernprogramme wäre eine weitgehende Vereinheitlichung dieser Oberflächenkonstituenten in Bezug auf Funktion, Form und Bildschirmposition wünschenswert.
Anhand des Übungsprogramms "Verbindungsarten" (Deutsch als Fremdsprache: Fachsprache Maschinenbau/Konstruktionslehre) werden Vorschläge zur Ausgestaltung dieser Status- und Steuerelemente vorgestellt.

In order to use computers as a medium to learn a language there is need for some GUI-elements indicating and regulating the actual status of training. These elements have to be separated in obligatory ones, useful in (almost) all CALL-software, and facultative ones, neccessary for most "teachware". Due to a general need of reducing the difficulties to get into the use of a training programm for language learning a standardization - as far as possible - of these GUI-elements should be aimed for.
Suggestions for design and function are made on the basis of the training programm "Verbindungsarten" (German as a foreign language/for special purposes: mechanical engineering).

1.	Relevanz des Themas	2
2.	Reduzierung des Einarbeitungsaufwands	3
2.1	Einarbeitungsaufwand und Lernprogramme	3
2.2	Allgemeine Gestaltungsgrundsätze	4
2.2.1	Vereinheitlichung	4
2.2.2	Analogiebildung und Transparenz	5
2.2.3	Farben, Formen, Töne	7
2.3	Chancen einer Standardisierung	7
3.	Obligatorische Steuerelemente	8
3.1	Aufteilung der Oberfläche	8
3.2	Autonomes Lernen und der *Ende*-Befehl	9
3.3	Hilfsmittel: Vokabelkartei, Wörterbuch, Grammatik	11
3.4	Üben und Testen	12
4.	Fakultative Steuerelemente	14
5.	Perspektive	17

FL022911

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Haymo Mitschian
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

BEST COPY AVAILABLE

1. Relevanz des Themas

In fast schon regelmäßigen Abständen erfährt man aus mehr oder weniger seriösen Publikationen von der demnächst anstehenden Ablösung des herkömmlichen Lernens in Schulen durch computergestützten Unterricht. Sieht man sich dagegen den Alltag des Lernens an, so findet man zumindest auf dem Gebiet des Fremdsprachenerwerbs wenig, was diese Ablösung in absehbarer Zeit als wahrscheinlich erscheinen läßt.

Denn im wesentlichen beschränkt sich das derzeitige Angebot an computergestützten Sprachlernmitteln auf elektronische Vokabelkarteien, Grammatiken oder Lexika, alles zwar wichtige Hilfsmittel, die den Spracherwerb stützend begleiten, ihn jedoch nicht tragen können. Vorstellungen, wonach Sprache hauptsächlich aus Wörtern und den Regeln ihrer Beziehungen zueinander bestehe und man eine Sprache lernen könne, indem man eine hinreiche Anzahl von Vokabeln und die wichtigsten Grammatikregeln dem Gedächtnis einpräge, sind definitiv veraltet. In modernen Sprachlehr- bzw. -lernmethodiken wird dem Vokabel- und Grammatiklernen eine zwar unentbehrliche, aber doch nur periphere Rolle zugewiesen. Im Zentrum stehen die Lernenden, denen durch abwechslungsreiche, interessante und dadurch motivierende Übungsangebote ein aktives Lernverhalten ermöglicht werden soll, das erwiesenermaßen die besten Lernerfolge erbringt.

Die Ursachen für die Diskrepanzen zwischen den Ansprüchen der Fremdsprachendidaktiken und der Praxis computergestützten Sprachlernens liegen auf verschiedenen Ebenen (vgl. Mitschian 1993:261ff.). Nach einer Phase relativer Offenheit gegenüber computergestütztem Lernen etwa Mitte der 80er Jahre machte sich an vielen Lehrinstitutionen eine Ernüchterung breit, zum Teil verbunden mit deutlichen Vorbehalten gegenüber maschinenbasiertem Lernen. Ausschlaggebend waren vor allem die Lücken zwischen Versprechungen und Erwartungen einerseits, geweckt durch Schlagwörter wie das von der *Künstlichen Intelligenz*, und andererseits den umständlich zu bedienenden, methodisch aber dennoch simplen Paukprogrammen, die tatsächlich zur Verfügung standen. Durch die schnelle Weiterentwicklung auf dem Hard- und Softwaremarkt wurde diese Situation nicht unbedingt nur verbessert, da viele Sprachlehrinstitute schon aus finanziellen Gründen nicht in der Lage sind, mit dieser Entwicklung Schritt zu halten.

Als Folge davon gibt es heute zwar reichlich Theorie zum computergestützten Fremdsprachenlernen, was jedoch fehlt, sind praktische Erfahrungen in größerem Umfang, die für eine fundierter Weiterentwicklung von Sprachlernsoftware unentbehrlich sind. Störend wirkt sich auch immer noch die Existenz einer Kontaktschwelle zwischen Didaktikern und Programmierern aus, die erst nach und nach überwunden wird. Gegenwärtig gibt es noch viele Programme, die entweder methodisch oder technisch gut sind, nicht jedoch beides zusammen.

Der Kreislauf, der sich auf diese Weise zwischen dem Fehlen qualitativ hochstehender Software und einem schmalen Absatzmarkt, der Investitionen in Programme erst ermöglicht, ergibt, ist zwar nicht hermetisch geschlossen, behindert die Entwicklung doch spürbar. Aufgelockert wird er vor allem durch die voranschreitende Verbreitung der Computer, die verstärkt auch in Lehrinstitutionen für die üblichen Zwecke der Datenverarbeitung vorhanden sind, und deshalb nicht mehr speziell für den Einsatz von Lernprogrammen angeschafft werden müssen. Produktionsfördernd dürfte sich auch die Verbreitung der graphischen Benutzeroberflächen auswirken, die die oft sehr umständlichen befehls- oder rein

menügesteuerten Programme wohl verdrängen werden und somit zu einer gewissen Vereinheitlichung und Beruhigung der Entwicklung beitragen.

2. Reduzierung des Einarbeitungsaufwands

2.1 Einarbeitungsaufwand und Lernprogramme

In einer Reduzierung von Vielfalt scheint generell ein Schlüssel sowohl zur Produktions- als auch zur Akzeptanzsteigerung speziell von computergestützten Sprachlernmaterialien zu liegen. Jede Software erfordert vom Benutzer einen gewissen Einarbeitungsaufwand, schon allein deshalb, weil jedes Programm in irgendeiner Weise Neues enthalten sollte, was der Benutzer erst kennenlernen muß, in das er sich also einzuarbeiten hat. Je höher nun aber der Einarbeitungsaufwand in Relation zum Gewinn ausfällt, den man sich von dem Programm verspricht, desto größer wird die Gefahr, daß sich potentielle Nutzer von der Arbeit damit abhalten lassen.

Der Umfang des Einarbeitungsaufwandes wird nun unter anderem davon bestimmt, wie viele singuläre bzw. übertragbare Bedienungselemente ein Programm enthält. Je mehr Elemente aus anderen Programmen bereits bekannt oder deren Kenntnis später wieder für andere Programme nützlich sind, desto geringer wird der absolute oder - im zweiten Fall - der relative Einarbeitungsaufwand. Wie hoch der relative Aufwand ausfallen darf, ist zusätzlich von einem Zeit- bzw. Intensitätsfaktor abhängig. Permanent und vielfältig nutzbare Software kann mit einem hohen Einarbeitungsaufwand verbunden sein, ohne dadurch benutzerunfreundlich zu werden. Je umfangreicher die Nutzung des Programms ausfällt, desto geringer werden im Verhältnis dazu die Anstrengungen beim Einstieg. Dies gilt beispielsweise für Textverarbeitungsprogramme für den Fall, daß die gesamten Schreibarbeiten in einem Bereich auf Computerbasis umgestellt werden. Die dauerhafte und häufige Nutzungsmöglichkeiten der Textverarbeitung relativieren eine gegebenenfalls auch etwas höhere Zugangsschwelle.

Genau aus diesem Grund besteht allerdings für Autoren von Lernprogrammen die Verpflichtung, den Einarbeitungsaufwand in ihre Produkte auf einem absolut niedrigen Niveau zu halten. Ein Lernprogramm dient dem Zweck, ein bestimmtes Wissen, bestimmte Fähigkeiten oder Fertigkeiten zu erwerben. Ist dieses Ziel erreicht, verliert das Programm seinen Wert für den Benutzer. Die Nutzungsdauer ist also vergleichsweise kurz, was den Lernenden nicht nur von Anfang an bekannt ist, sondern zudem in ihrem Interesse liegt, da sie normalerweise in möglichst kurzer Zeit möglichst viel Lernstoff bewältigen wollen. Zu bedenken ist außerdem, daß bei Lernprogrammen die Einarbeitung Leistungen erfordert, die anfallen, lange bevor man zum inhaltlichen Lernen kommt, was selbst wieder nur eine Vorstufe zum eigentlichen Tätigkeitsziel ist, nämlich dem Beherrschen und der Anwendung des Gelernten. *Einarbeitung* bedeutet also im Kontext eines Lernprogramms zu lernen, wie man ein Hilfsmittel einsetzt, mit dessen Unterstützung man etwas lernen kann, von dessen Erwerb man sich in naher oder ferner Zukunft Nutzen verspricht. Schon deshalb muß bei Benutzern reiner Lernprogramme von einer nur geringen Einarbeitungsbereitschaft ausgegangen werden.

Beim Fremdsprachenlernen am Computer können noch weitere Stufen zwischen Einarbeitung und Lernziel hinzukommen, nämlich dann, wenn metasprachliche Beschreibungen als

Lernhilfen eingeschaltet werden. Von den Sprachwissenschaftlern unter den Fremdsprachenlernern einmal abgesehen werden zum Beispiel Grammatik- oder Wortbildungsregeln nicht aus Interesse an den Sprachstrukturen selbst gelernt, sondern weil Beschreibungssysteme dieser Art zumindest erwachsenen Lernern Hilfen anbieten. Ist das übergeordnete Lernziel *Sprachbeherrschung* bis zum gewünschten Grad erreicht, darf das explizite Regelwissen sogar wieder vergessen werden, da es dann in Form internalisierter Fertigkeiten zur Verfügung steht. Die Einarbeitung in das Programm liegt hier also weit vor dem eigentlichen Lernziel.

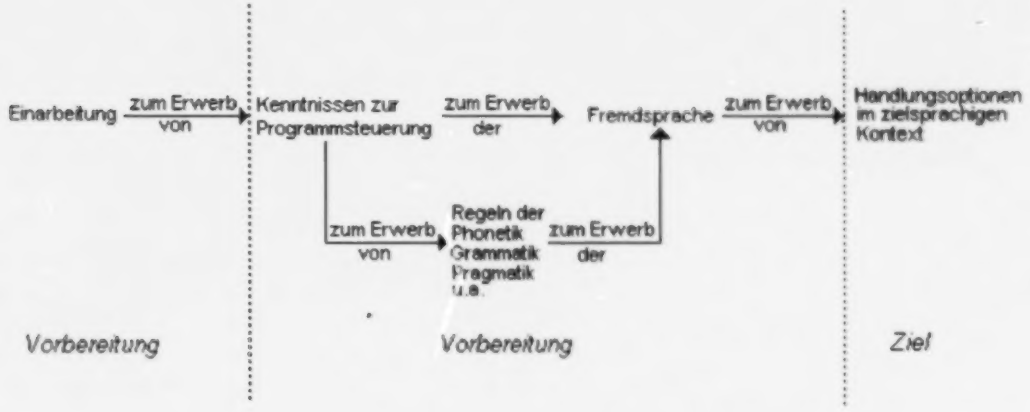

Abb. 1: Lernebenen beim computerunterstützten Fremdsprachenerwerb

2.2 Allgemeine Gestaltungsgrundsätze

Zur Reduzierung des Einarbeitungsaufwandes stehen zunächst die Möglichkeiten der Softwareergonomie offen, die sich im Kern in vier Grundsätze fassen lassen:

1. Gleiches gleich behandeln;
2. nach Anknüpfungspunkten zum Erfahrungshintergrund der Anwenderzielgruppe suchen, um darüber einen intuitiven Zugang zu den Programmfunktionen zu eröffnen;
3. komplexe und damit schwer durchschaubare Programmstrukturen nach Möglichkeit vermeiden;
4. sorgfältig bedachter und stets motivierter Einsatz graphischer und farblicher Elemente und ein ebensolcher Umgang mit akustischen und visuellen Animationen.

2.2.1 Vereinheitlichung

Der Grundsatz, Gleiches gleich zu behandeln, hat sich nicht nur für den Bereich der Gestaltung und Funktionszuweisung von Oberflächenelementen innerhalb eines Programms durchgesetzt, sondern sorgt sogar für Konsistenzen in der Bedienung von Programmen verschiedener Softwarehersteller. Die sich hier abzeichnende Normierung wirkt sich eindeutig positiv in Bezug auf den Einarbeitungsaufwand aus. Gestaltet man deshalb Lernprogramme nach den von führenden Softwareproduzenten vorgelegten Richtlinien zum

BEST COPY AVAILABLE

Anwendungsdesign, dann reduziert sich die Einarbeitung bzw. wird transferierbar, was mitunter als zusätzliches Lernziel oder zusätzliche Motivation integriert werden kann. Auch wenn es schwer fällt, sollte deshalb auf das kreative Entwerfen neuer Steuerelemente verzichtet. Denn diese Kreativität führt zunächst immer zu singulären Lösungen, die nur dann gerechtfertigt sind, wenn Vorhandenes zur Bewältigung der gestellten Aufgaben nicht ausreicht.

Der Druck zur Vereinheitlichung beschränkt sich jedoch nicht nur auf Form und Funktion von Oberflächenelementen. Das übergeordnete Ziel der Oberflächengestaltung ist es, dem Benutzer die Arbeit mit dem und die Orientierung im Programm zu erleichtern. Dabei fällt auch der Anordnung der Elemente eine determinierende Rolle zu. Anders als jedoch bei Entwurf und Funktionszuweisung von Steuerelementen, was zum Teil sehr arbeitsintensiv sein kann und schon deshalb unterbleibt, stellt die Verteilung der Elemente auf dem Bildschirm keine besonderen Anforderungen an das Können der Programmierer bzw. Lernsystemanalytiker. Dennoch sollte dieser Freiraum nicht unbedingt genutzt, sondern auch hier auf größtmögliche Konsistenz über die Programmgrenzen hinweg geachtet werden.

Dies alles soll nicht in eine willkürliche Unterdrückung individueller Kreativität münden. Das ausschlaggebende Kriterium ist jeweils die Begründbarkeit einer Maßnahme. Anhaltspunkte für eine Begründung liefern zunächst und absolut die technisch-physikalischen Bedingungen der Oberflächengestaltung. Als nächstes sind softwareergonomische und wahrnehmungs- oder lernpsychologische Überlegungen anzustellen, die meist untrennbar mit Fragen der ästhetischen Gestaltung verknüpft sind. Aber schließlich kommt eben auch das Argument der Vereinheitlichung hinzu und damit der Vorbildfunktion kommerziell erfolgreicher Softwareprodukte, die allein schon durch ihre Verbreitung prägend wirken.

So finden sich zum Beispiel in sprachbezogenen Lernprogrammen Menüs an allen möglichen Stellen, sei es der obere oder untere Bildrand, am linken oder rechten, zentriert, frei beweglich in einem eigenen Kontrollelement oder auf einer ganzen Bildschirmseite verteilt. Es gibt nur wenige technische, psychologische oder ergonomische Faktoren, die eindeutig für oder gegen eine bestimmte Positionierung eines Menüs sprechen. Von daher existiert aber auch kaum ein Grund, von der sich offensichtlich etablierenden Anordnung am oberen Rand unter einer Titelzeile abzuweichen.

2.2.2 Analogiebildung und Transparenz

Ein anderes bewährtes Mittel, um Programmabläufe transparent zu gestalten und intuitiv zugänglich zu machen, sind bekanntlich Analogiebildungen. Im Rahmen der softwaretechnischen Möglichkeiten werden Gegenstände oder Vorgänge aus dem Erfahrungsbereich der Benutzer nachgeformt oder symbolhaft auf dem Bildschirm dargestellt. Sie sind dann auf ähnliche Weise handzuhaben wie ihre realen Vorbilder. Bekanntestes Beispiel dürfte der "Papierkorb" sein, der in Textverarbeitungsprogrammen zur Zwischenablage oder zum Entfernen nicht benötigter Textteile dient. Nach demselben Prinzip sind die Dateimanager aufgebaut, die Dateien in Verzeichnissen und Unterverzeichnissen ablegen, oder Datenbankabfragen in Form von Karteikästen.

Die Ansatzpunkte zur Analogiebildung, die sich für fremdsprachliche Lernprogramme anbieten, ergeben sich zunächst aus dem Vergleich mit herkömmlichen Lernmaterialien, also

in erster Linie mit papierbezogenen Lernhilfen wie Büchern oder Arbeitsblättern. Bei multimedial angelegter Lernsoftware ergibt sich ein Bezug zu den Lehrwerken, also Büchern kombiniert mit Bild- und Tonmaterialien auf unterschiedlichen Trägern. Das Ziel der Analogiebildung sollte nun nicht sein, Bücher auf dem Bildschirm nachzubilden. Dadurch würden spezifische Vorteile des Lernmediums Computer verschenkt werden. Wichtig ist es, vorhandene, aus der Arbeit mit Papier entstandene Lerntechniken und -strategien zu erkennen, und dann für die Bildschirmarbeit entweder Ansatzpunkte für das gewohnte Lernverhalten anzubieten oder das Lernverhalten gezielt zu verändern suchen, indem bessere Lernwege eröffnet werden.

Ein banales, aber gleichzeitig anschauliches Beispiel bieten die Orientierungstechniken bei der Arbeit mit einem Buch. Schlägt man eine Seite auf, sieht man immer auch, wieviele Seiten vor und nach der aktuellen Stelle vorhanden sind, beim seriellen Lernen mit einem Buch also ungefähr wieviel des Lernstoffes bereits bearbeitet ist und wieviel noch vor einem liegt. In ähnlicher Wiese liefern teils automatisierte Orientierungstechniken Informationen zum gegenwärtigen Stand innerhalb von Kapiteln, Abschnitte bis hinein in einzelne Übungen und Übungsaufgaben. Lernende erhalten dadurch permanent Hinweise darauf, was sie bereits geleistet haben, wie umfangreich der aktuelle Lernabschnitt insgesamt ist und wieviel Zeit und/oder Konzentration von ihnen noch bis zum nächsten Einschnitt verlangt wird.

Anders dagegen verhält es sich bei der Arbeit mit Bildschirmseiten. Bereits bearbeitete oder noch zu bearbeitende Seiten sind nicht unmittelbar sichtbar, materiell auch nicht vorhanden. Entweder kann man überhaupt nicht erkennen, auf welcher "Seite" des Programms man sich gerade befindet, oder aber die angebotenen Orientierungshilfen erfüllen ihren Zweck nicht, weil sie ungewohnt, nicht deutlich genug oder einfach nicht ausreichend sind. Die fehlende Orientierung erschwert den Lernenden die Organisation ihrer Lerntätigkeit, da sie nicht erkennen können, wo Pausen auftreten, wie lange Phasen der Konzentration dauern oder wie hoch der Aufwand zur Bewältigung eines Lernschrittes sein wird.

Da Software gerade nicht an die physikalischen Einschränkungen des Buchs gebunden ist, eröffnen sich neue Wege der Informationspräsentation, die dann jedoch auch Analogien stören, vielleicht sogar zerstören können. Inhaltlich-funktionale Verbindungen lassen sich so nach Art der Hypersysteme ohne Umwege herstellen, mit direkten Zugriffsmöglichkeiten von einem Begriff auf seine Definition, ein erläuterndes Bild und Verweise auf weitere Informationsquellen. Die in einem Buch vom Inhaltsverzeichnis oder anderen Registern wahrgenommenen Aufgaben können in eine Programmoberfläche integriert werden, so daß sich die Bearbeitung auf wechselnden Informationsebenen fortführen läßt. Diese schnellen, komplikationslosen Sprünge innerhalb des Systems bergen jedoch auch das Risiko des Orientierungsverlustes in sich, bekannt unter dem Schlagwort *lost in hyperspace*.

Zwingender noch als in anderer Software sollten also in Lernprogrammen permanente Orientierungsanzeigen vorhanden sein. Komplexe Programmstrukturen, die ständige Wechsel zwischen Ebenen erfordern, sollten generell vermieden werden, vor allem dann, wenn sie nicht mehr in eine Analogie integrierbar sind. Ein wesentliches Orientierungsmittel auf graphischen Betriebssystemen ist eine weitgehend stabile Oberfläche mit Elementen, die im gesamten Programmverlauf vorhanden bleiben, anstelle eines Bildschirmaufbaus, der nach jedem Klick neu erfolgt. Zur Ausstattung von Übungen sollte auf jeden Fall eine Positions- und Umfangsangabe gehören, dazu ständig aktualisierte Rückmeldungen zum Übungsstand,

eventuell auch kommentiert, sowie ein Zugang zur Programmsteuerung und zu Zusatzinformationen.

2.2.3 Farben, Formen, Töne

Zu den Stärken des Computers zählen die Flexibilität bei der graphischen Gestaltung, die technisch fast uneingeschränkte Einsetzbarkeit von Farben sowie die Rechnerschnelligkeit, die die Darstellung komplexer Animationen erlaubt. Durch die Erweiterung auf Multimedia kommen weitere akustische und visuelle Möglichkeiten hinzu. In diesen Optionen liegt natürlich auch die Gefahr der Übertreibung.

Wie schon bei den Fragen der Gestaltung und Positionierung der Steuerelemente sollte hinter dem Einsatz aller akustischer und visueller Hilfsmittel ihre Funktionalität stehen. Markierende, hervorhebende Farben gehen auf einer bunten Oberfläche ebenso verloren wie akustische Hinweise in einer permanenten Klangberieselung. Bei Lernprogrammen muß mehr als bei anderer Software damit gerechnet werden, daß sie häufig an Terminals in gemeinschaftlich genutzten Räumen schulischer Einrichtungen eingesetzt werden, wodurch sich eine nicht steuerbare, v.a. nicht abschaltbare Beschallung von selbst verbietet. Starke Farbkontraste oder Blinken, an sich bewährte Optionen zur Steuerung der Aufmerksamkeit, beanspruchen die Augen der Benutzer zusätzlich, die bei überwiegend textbasierter Bildschirmarbeit bereits stark belastet werden. Der Hinweis auf eine sparsame und geordnete Verwendung dieser Gestaltungselemente, der für alle Programme gilt, besitzt bei Lernsoftware mit ihrer erhöhten Wiederholungsfrequenz durch das Üben besondere Relevanz.

Soweit sie nicht selbst Träger von Informationen sind, können Farben, Formen, Bilder und Töne zur nonverbalen Steuerung der Benutzer eingesetzt werden, wenngleich oft nur in Kombination mit einigen oder anderen Elementen. Diese gerade im Kontext des Fremdsprachenlernens wichtige Funktion läßt sich nur auf der Basis vorhergehender Übereinkunft realisieren. Allgemeine Konotationen, wie zum Beispiel bei Farben, können kulturell unterschiedlich oder zu pauschal gehalten sein, um eindeutig eingesetzt werden zu können. Die meisten Vorteile dürften auch hier wieder über eine weitgehende Anpassung an Vorhandenes erzielbar sein, also nach dem Grundsatz Gleiches gleich, und dies über Programmgrenzen hinweg, um einen Wiedererkennungseffekt nutzen zu können, der unter Umständen weitausholende Erklärungen erübrigt.

2.3 Chancen einer Standardisierung

Dies alles läuft auf eine weitgehende Standardisierung von Lernprogrammen bezogen auf ihre Elemente und Darstellungsweisen hinaus. Eine Normierung im eigentlichen Sinne, um den Programmen ein einheitliches Outfit zu verpassen, ist jedoch weder möglich noch wünschenswert. Doch zum einen determinieren vorhandene Oberflächenelemente den Gestaltungsspielraum, schon allein deshalb, weil sich durch ihre Verwendung der Programmieraufwand erheblich reduzieren läßt. Und zum anderen führt zumindest bei durchdachtem Vorgehen kein Weg daran vorbei, für Form und Funktion eines jeden Oberflächenelements Begründungen bereitzuhalten, die sich aus der gestellten Aufgabe herleiten.

Im Rahmen der technischen Möglichkeiten liefert zunächst die sprachlernspezifische Art der Aufgabenstellung solche Begründungen. Bevor nun aber eigene Steuerelemente genau für diese Zwecke erfunden werden, sollten Programmierer versuchen, mit Bekanntem zurechtzukommen. Mittlerweile existiert ein breites Spektrum von Steuerelementen, die für eine Vielzahl denkbarer Aufgabenstellungen ausreichen müßte. Des weiteren geben die erwähnten Analogiebildungen einen geeigneten Weg ab, um Vorgehensweisen zu begründen. Da der Einsatz unterschiedlicher Medien fester Bestandteil der Gestaltung des Fremdsprachenerwerbs ist, beschränken sich Anknüpfungspunkte für Analogien nicht nur auf Bücher oder andere papierbezogene Informationsträger.

Neben ihrer Funktion, für einen motivierten Einsatz von Oberflächenelementen zu sorgen, eignen sich diese Begründungen auch als Diskussionsgrundlagen zur Bewertung der Oberflächengestaltung und können als Basis der Weiterentwicklung dienen. Diese Funktion ist auch den nachfolgenden Vorschlägen zugedacht, wobei noch differenziert werden kann zwischen obligatorischen Steuerelementen, die für alle fremdsprachenbezogenen Lernprogramme notwendig erscheinen, und fakultativen, für viele Programme nützliche.

3. Obligatorische Steuerelemente

Um von der reinen Theorie wegzukommen, sollen die Vorschläge an einem Beispiel demonstriert werden, damit gleichzeitig deren Umsetzbarkeit belegt werden kann. Das Beispiel ist ein an der TU Berlin am *Institut für Fachdidaktik Deutsch und Fremdsprachen* entstandenes bzw. noch entstehendes Deutsch-Lernprogramm für ausländische Studenten aus dem Bereich Maschinenbau/Konstruktionslehre. Es handelt sich dabei um Übungsmodule, die mit verschiedenen Inhalten gefüllt werden können. Hier sind dies die Verbindungsarten *Kleben, Löten, Nieten, Schrauben* und *Schweißen*.

3.1 Aufteilung der Oberfläche

Die Oberfläche eines Lernprogramms muß drei unterschiedliche Bedarfsarten berücksichtigen. Der vermutlich immer größte Teil dient als Arbeitsfläche, auf der die eigentliche Informationsvermittlung bzw. Übungstätigkeit stattfindet. Daneben müssen Elemente angezeigt werden, über die der Programmablauf beeinflußbar ist, also die Steuerelemente im eigentlichen Sinn. Schließlich sollte der Benutzer über diesen Ablauf und den Stand seiner Lerntätigkeit auch informiert werden, wozu einige Elemente zur Anzeige des Programmstatus nötig sind. Sinnvollerweise sollten die Steuer- und Statuselemente gebündelt dargestellt werden, um schon so gegenseitig Hinweise auf ihre Funktionalität zu liefern.

In älteren Programmen war es aus Gründen der Platzersparnis durchaus üblich, diese drei Elementgruppen auf eigene Bildschirmseiten zu verteilen. Die Nachteile dieser Anordnung liegen auf der Hand: das Programm wird komplexer, umständlicher zu bedienen und insgesamt schwerer durchschaubar. Das Auflösungsvermögen der Monitore erlaubt heute einen etwas großzügigeren Umgang mit der vorhandenen Fläche. So ist zum Beispiel schon bei einer Auflösung von 640 zu 480 Punkten auf einer Bildschirmseite mehr Text darzustellen, als normalerweise in Lernprogrammen sinnvoll sein dürfte. Denn wenn es darum geht, einen längeren Text nur zu lesen, dann eignet sich ein Buch oder ein Ausdruck

immer noch besser als der Bildschirm. Der Platzbedarf der Steuer- und Statuselemente verursacht daher keinen wirklichen Verlust an Fläche, wenn sie platzsparend an den Rändern positioniert werden. Falls wirklich nötig, können sie immer noch ausblendbar angelegt werden.

Wie schon für die Menüzeile existiert auch für diese Elementgruppen kein Argument, das für einen bestimmten Bildschirmrand spräche. Lediglich soweit sie Schrift enthalten ist eine horizontale Anordnung am oberen oder unteren Rand sinnvoll, da sich eine vertikale Reihung in der Breite immer am längsten Eintrag zu richten hat, wodurch Platz verschenkt würde. Für eine Leiste mit Steuerelementen am oberen und für Statusanzeigen am unteren Bildrand spricht dagegen die Oberflächengestaltung anderer Programme, z.B. vieler Textverarbeitungssysteme. Auch die Anbindung an die Menüzeile, die ebenfalls der Programmsteuerung dient, legt den oberen Rand als die geeignete Stelle für Steuerelemente nahe, und zur besseren Abhebung davon dann den unteren Rand für die Statusanzeigen.

In diesen beiden Gruppen sollten diejenigen Schalter und Anzeigen gesammelt und permanent sichtbar sein, die für den gesamten oder zumindest für weite Teile des Programmablaufs von Bedeutung sind. Ein ständiges Ein- und Ausblenden von Schaltern oder Feldern kann die Orientierung im Programm erschweren, während stabile, immer vorhandene Elemente den gegenteiligen Effekt bewirken. Soweit sie für einen bestimmten Programmstatus irrelevant sind, ist es besser, diese Elemente zu deaktivieren und diesen Zustand durch Kontrastreduzierung anzuzeigen.

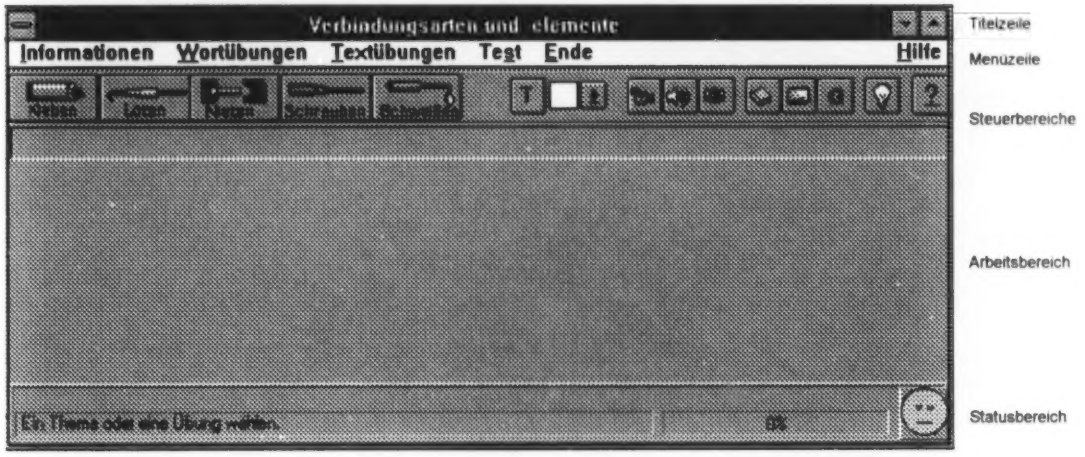

Abb. 2: Oberflächenaufteilung nach Bereichen

3.2 Autonomes Lernen und der *Ende*-Befehl

Computerunterstütztes Lernen wird häufig im Zusammenhang mit autonomen, also selbstbestimmten Lernformen gesehen. Um selbst bestimmen zu können, müssen die Lernenden jedoch wissen, welche Optionen ihnen überhaupt zur Verfügung stehen. Deshalb muß in Lernprogrammen ein kursorischer Überblick über alle Teile des Programms einschließlich ihrer Funktionen möglich sein. Testläufe mit Lernsoftware haben gezeigt, daß viele Lernende erst einmal versuchen, sich durch das Programm zu "zappen", um zu sehen, ob das vorhandene Angebot an Informationen und Übungen oder zumindest Teile davon

überhaupt mit ihren Lernintentionen übereinstimmen. Kommt es dabei zu Austiegsproblemen aus Programmteilen, sind Hemmungen beim weiteren Erkundungsversuchen nicht selten zu beobachten, wenn nicht sogar dadurch schon das Interesse an einer Arbeit mit dem Programm verschwindet.

Zum autonomen Lernen gehört es auch, daß die Lernenden Beginn, Dauer und Intensität der Lerntätigkeit selbst festlegen. Während es nur selten Schwierigkeiten macht, ein Programm oder Teile davon zu starten, und die Intensität der Lerntätigkeit von der Konzentrationsfähigkeit der Lernenden und der Motivationskraft des Lernangebots abhängt, sind die Schwierigkeiten, die beim Beenden auftreten können, vielen nicht nur aus der Fachliteratur zum computerunterstützten Lernen bekannt (Brücher 1991:177).

Die Steuerelemente, mit denen ein Programm ordentlich beendet werden könnte, sind häufig nicht auf allen Programmebenen zugänglich. Für den Benutzer bedeutet dies, daß er zum Beenden erst auf bestimmte Ebenen wechseln muß, wozu unter Umständen eine ganze Sequenz von Bedienungsschritten notwendig ist. Ebenfalls nicht selten werden die *Ende*-Befehle in Untermenüs versteckt, wo sie erst nach längerem Suchen entdeckt werden. Der Ausstieg aus dem Programm wird auch nicht erleichtert, wenn er jederzeit über bestimmte Tasten oder Tastenkombinationen möglich ist, welche Tasten zu drücken sind aber nicht auf dem Bildschirm zu erfahren ist Verschärft wird diese Schwierigkeit noch bei einer wechselnden Tastenbelegung.

In manchen Programmen ist der *Ende*-Befehl nicht nur zufällig oder aus Versehen nicht aktivierbar, sondern er wird aus pädagogischen Überlegungen gesperrt oder der Zugang dazu zumindest erschwert. Die Lernenden sollen dadurch zu bestimmten Übungsschritten gezwungen werden, bevor sie die Arbeit beenden dürfen. Dies widerspricht nicht nur dem Konzept des autonomen Lernens, das auf einer starken Eigenverantwortlichkeit der Lernenden basiert. Die Möglichkeiten, sich vor Beginn der Lerntätigkeit einen Überblick über das Angebot zu verschaffen, wird dadurch eingeschränkt, wenn nicht sogar vollständig verhindert. Außerdem hat ein sinnvoller Einsatz derartiger pädagogischer Maßnahmen zur Bedingung, daß Lernvoraussetzungen, Lernstand und Lernprobleme der Programmnutzer eindeutig bestimmt werden könnten. Gerade dies ist jedoch beim computerunterstützten Lernen mit seinem in der Regel anonymen Verhältnis zwischen Autoren und Rezipienten nicht der Fall.

Die Wirkung, die man sich von dieser Maßnahme vermutlich verspricht, nämlich durch den Zwang zur Weiterarbeit den Lernenden dabei zu helfen, sich selbst zu disziplinieren und über einen längeren Zeitraum hinweg zu motivieren, läßt sich auch auf andere Weise erreichen, z.B. durch explizite Testphasen (s.u. Üben und Testen). Außerdem gibt es immer Wege, die Arbeit trotz aller eingebauter Hindernisse abzubrechen, und sei es mit Hilfe des Netzsteckers. Und schließlich können auch andere Gründe, etwa technische, dazu nötigen, ein Programm zu einem beliebigen Zeitpunkt verlassen zu müssen.

Somit ergeben sich einige Anforderungen, die an das Steuerelement zur Beendigung gestellt werden: es muß permanent zugänglich, sichtbar und aktivierbar sein, darf also nicht etwa in einem Untermenü versteckt werden. Außerdem muß es differenziert reagieren, je nachdem was beendet werden soll. Dies kann lediglich die Übungsaufgabe sein, die gerade zur Bearbeitung ansteht, oder aber es soll von einer Übung zu einer anderen gewechselt werden, und schließlich das Programm selbst. Von daher empfiehlt sich ein dreiteiliger Aufbau:

1. Beenden des aktuellen Übungsschrittes,
2. Beenden der aktuellen Übung,
3. Beenden des Programms.

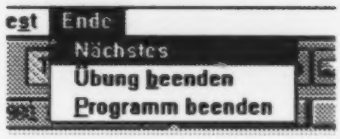

Abb. 3: Aufbau des *Ende*-Menüs

Erfüllt werden diese Anforderung zum Beispiel durch einen Schalter in der Steuerleiste, der ein Dialogfenster mit den drei genannten Möglichkeiten aufruft oder, wie im Lernprogramm *Verbindungsarten*, durch einen Eintrag in der Menüzeile mit den entsprechenden Menüpunkten. Setzt man das *Ende*-Menü an den rechten Rand der Menüzeile, erschließt man dadurch einen zusätzlichen Steuerungsimpuls. Die von vornherein vorhandene Tendenz, schriftlich angezeigte Menüs von links nach rechts abzuarbeiten, wird durch diese Anordnung des *Ende*-Menüs verstärkt, was eine Beeinflussung von Lernvorgängen ermöglicht.

3.3 Hilfsmittel: Vokabelkartei, Wörterbuch, Grammatik

Wie bereits eingangs erwähnt, sind Vokabelkarteien, Wörterbücher oder Grammatiken unentbehrliche Hilfsmittel des Fremdsprachenlernens. Beim Sprachlernen fällt eine Menge arbiträren Lernmaterials an, also Lernstoff, der nicht irgendwie logisch erfaßt werden kann, sondern schlichtweg auswendig gelernt werden muß. Dies trifft fast auf das gesamte Vokabular und auf viele Wort- und Satzbildungsregeln zu. Es gibt zwar einen sprachhistorischen, jedoch keinen aus der heutigen Sprachform ableitbaren Grund, weshalb der Gegenstand *Stuhl* im Deutschen so, im Englischen *chair* und im Chinesischen *yizi* benannt wird. Es handelt sich also hierbei um Lerngegenstände, die nicht begreifend-verstehend erfaßt werden können, sondern primär über ein wiederholendes Üben zu erwerben sind, wozu Sammlungen, Listen oder sonstige Übersichten eine Grundlage bilden.

Was an Computerarbeitsplätzen meistens fehlt, ist Raum für zusätzliche Lernhilfen. Tastatur und Maus liegen normalerweise genau auf der Fläche, die man zum Schreiben und Lesen benötigen würde. Zudem ist es umständlich und anstrengend, von der Tastatur zum Kugelschreiber, vom Lesen am Bildschirm zum Lesen vom Blatt zu wechseln. Die Fenstertechnik graphischer Benutzeroberflächen ist dagegen extra für die gleichzeitige Arbeit mit verschiedenen Gegenständen konzipiert. Es stellt also technisch kein Problem dar, ein Fenster für den Zugriff auf eines oder alle der genannten Hilfsmittel zur Verfügung zu stellen. Wenn trotzdem nicht alle gleichzeitig aktivierbar sind, so hat dies oft mit Fragen der Übersichtlichkeit oder schlicht des Platzbedarfs zu tun.

Es ist auch kein Zufall, weshalb die Mehrzahl der vorhandenen Software zum Spracherwerb Nachschlage- oder Trainingsprogramme für Vokabeln oder Sprachregeln sind. Hier können die altbekannten Stärken der elektronischen Datenverarbeitung ausgespielt und umfangreiche Such-, Sortier- oder Auswahlroutinen sinnvoll angewandt werden. Vor allem die Techniken der hypermedialen Textaufbereitung dürften noch ein erhebliches Potential in sich bergen. Von daher läge es nahe, Sprachlernprogramme für den Einsatz vorhandener Sprachlernmittel zu öffnen, also nicht jedes Lernprogramm mit einer eigenen Vokabelkartei und einem eigenen Wörterbuch auszustatten. Eingeschränkt gilt dies auch für Grammatiken, obwohl hier in vielen Fällen wohl eine Vorauswahl günstig wäre, um den Umfang überschaubar zu halten. Wie hoch auf diesem Gebiet die Chancen zu einer Vereinheitlichung bzw. Anpassung an allgemeine Standards ist, läßt sich schwer abschätzen. Auf jeden Fall wird es vorerst notwendig sein, jedes Sprachlernprogramm noch mit eigenen Hilfsmitteln auszustatten.

Wie schon der *Ende*-Befehl müssen auch sie ständig aktivierbar sein, schließlich ist nicht vorhersehbar, wer wann welche Hilfestellung benötigt. Ihrem Charakter entsprechend könnten sie im sicherlich vorhandenen Hilfemenü untergebracht werden. Um einen schnellen Zugriff zu erlauben, eignen sich auch Schalter für diesen Zweck. Eventuelle Auswahloptionen, z.B. zwischen verschiedenen Lexika oder Karteien, sind gut in einem Untermenü aufgehoben.

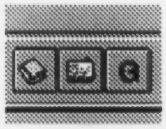

Abb. 4: Schalter für Wörterbuch, Vokabelkartei und Grammatik

3.4 Üben und Testen

Zu den wesentlichen Stärken des computerunterstützten Lernens zählt die Möglichkeit, die Eingaben der Lernenden durch Rückmeldungen zu bewerten und/oder zu kommentieren. Anders als beim personenbetreuten Lernen existieren keine Einschränkungen in Bezug auf die Geduld oder die Verfügbarkeit der Helfer. Außerdem sind die Rückmeldungen, da sie nicht adhoc gegeben werden müssen, mehrfach überprüft und dadurch verläßlicher. Ein Nachteil liegt in der im Vergleich zu den von Lehrern oder Tutoren gegebenen Hinweisen in der geringen Flexibilität der Programmeldungen, da nur antizipierte Eingaben differenziert, alle anderen mehr oder weniger nur pauschal behandelt werden können.

Ein weiterer Vorteil liegt in der Kürze der Zeit, die zwischen Leistung und Rückmeldung liegt. Es ist seit langem bewiesen, daß Korrekturen, Bewertungen oder andere Formen der Leistungskommentierung nur dann ihre lernhelfende Wirkung voll entfalten, wenn der zeitliche Abstand zwischen Aktion und Reaktion sich in einem bestimmten, meist sehr engen Rahmen bewegt. Dies gilt umso mehr für kleinschrittige Übungsaufgaben, wie etwa das Ausfüllen eines Lückentextes oder das Lösen von Multiple-Choice-Aufgaben, die möglichst umgehend bewertet werden sollten. Im Lernalltag an schulischen Einrichtungen liegen oft Tage oder sogar Wochen zwischen Leistung und Rückmeldung.

Daraus folgt für die Konzeption von Lernprogrammen nicht, daß jeder Tastendruck oder Mausklick sofort kommentiert werden müßte. Dies kann so sein, kann aber auch erst nach definitiven, explizit bestätigten Eingaben oder am Ende einer Übungssequenz erfolgen. Entscheidend ist, daß die Bewertungshinweise innerhalb der aktuellen Arbeits- genauer gesagt Konzentrationsphase gegeben werden.

Eine weitere und ohne größeren Aufwand realisierbare Option ist die Speicherung und statistische Aufbereitung aller Übungsschritte und -ergebnisse. Sofern sie auf diese Weise erfaßbar sind, lassen sich damit kurz- und langfristige Leistungsentwicklungen anzeigen, die motivierend wirken und Hinweise auf die weitere Lernweggestaltung enthalten können. Entscheidungen können auf dieser Basis vom Lernenden selbst getroffen werden oder aber das Programm schlägt einen Übungsverlauf vor.

Diese beiden Optionen, also die sofortige Bewertung von Leistungen und ihre Speicherung bzw. Darstellung, bergen aber auch die Gefahr in sich, daß sich die Lernenden ständig

beobachtet oder sogar überwacht fühlen. Dies liefe einem anderen Vorteil des computerunterstützten Lernens entgegen, nämlich der Möglichkeit zu einem unbeobachteten und dadurch von sozialem Druck befreiten Üben. Die oft im personenbetreuten Lernen auftretenden Hemmungen vor Fehlern, aus denen man bekanntlich lernt, könnte durch einen permanent zugeschalteten Kontrollmechanismus unnötigerweise auf die Arbeit am Computer übertragen werden. Das Üben sollte somit freigehalten sein von allen Kontrollen und deshalb auch von der Sammlung von Leistungsdaten.

Eine die Übungstätigkeit permanent begleitende Datensammlung, auf die die Lernenden keinen oder nur geringen Einfluß erhalten, ist folglich keine akzeptable Lösung. Andererseits wirken fehlerfreie oder fehlerarme Leistungen oft dann erst zusätzlich motivierend, wenn sie sich selbst oder anderen gegenüber belegt werden können, wozu sie allerdings vorher gesammelt werden müssen. Was benötigt wird, ist also eine zu- und natürlich auch abschaltbare Ergebnisspeicherung, die man erreicht, wenn man strikt zwischen reinem Üben und expliziten Testphasen unterscheidet.

Vom Verlauf, den Anforderungen oder der Tätigkeit her, besteht im Prinzip kein Unterschied zwischen Üben und Testen. Normalerweise wird jedoch das erstere mehr erklärende Teile oder Zugangsmöglichkeiten zu Informationen bereithalten und die differenzierten Formen der Rückmeldungen anbieten. Der entscheidende Unterschied zwischen den beiden Arbeitsformen liegt in der Limitierung, die bei den Testvorgängen in irgendeiner Weise vorhanden ist. Üblich sind zeitliche Einschränkungen, die Sperrung des Zugangs zu Hilfsmittel oder zu allen zusätzlichen Informationsquellen, reduzierte oder bis zum Ende einer Phase völlig ausbleibende Rückmeldungen und eben die Überwachung der Lerntätigkeit in der ein oder anderen Form mit den entsprechenden Ergebnisanzeigen. Immer wieder erstaunlich ist die Motivationskraft, die von verteilten Punkten und der Möglichkeit zum Leistungsvergleich mit anderen ausgeht.

Soweit realisierbar sollten deshalb zu allen Übungen Testsequenzen angeboten werden, in denen dann auch mal der spielerische Charakter oder eine anderweitig aufgelockerte Form der Informationspräsentation dominieren kann. Im Lernprogramm *Verbindungsarten* kann über einen Testschalter zwischen den beiden Modi hin- und hergeschaltet werden, außerdem läßt sich in vielen Fällen ein Schwierigkeitsgrad über eine Liste im Steuerteil einstellen. Parallel dazu können diese Einstellungen auch über ein eigenes Testmenü vorgenommen werden, das leichter als die graphischen Elemente der Oberfläche über die Tastatur zu erreichen ist.

Je nach Art des Testaufbaus müssen einige Ablaufanzeigen vorhanden sein. Der Umfang der zu lösenden Aufgaben sollte erkennbar sein, bei Zeitlimits der jeweils aktuelle Stand oder zumindest motivations- und leistungsfördernde Zwischenergebnisanzeigen. Ein den Statusanzeigen oder -meldungen reservierter Teil der Oberfläche eignet sich dann auch besonders für die Aufnahme dieser Elemente, da dann schon die Anordnung zutreffende Rückschlüsse auf die Funktion erlaubt.

Als günstig für die reinen Übungssequenzen hat sich eine einfache graphische Rückmeldung erwiesen, die knapp über den Effekt der letzten Eingabe informiert. Nötig sind dazu drei, besser noch vier Anzeigen, die für ein negatives, neutrales, positives und - zum Beispiel für das erfolgreiche Beenden eines Übungsabschnittes - "besonders positives" Feedback.

Abb. 5: Beispiele für Rückmeldungszeichen

4. Fakultative Steuerelemente

Die Trennlinie zwischen Steuerelementen, die obligatorische Bestandteile aller fremdsprachenbezogener Lernprogramme sein sollten, und denjenigen, die fakultativ eingesetzt werden können, ist natürlich weder scharf noch absolut zu ziehen. Lernprogramme können etwa zur Einübung eines hochspeziellen einzelsprachlichen Phänomens dienen, das ganz spezifische Zugangsweisen erfordert. Oder aber es werden Lernspiele entwickelt, die sich von vornherein in kein Schema zwängen lassen.

Wie jedoch auch auf dem Lehrbuchmarkt kann bei Lernsoftware zwischen kurstragenden und -ergänzenden Lernmitteln unterschieden werden. Kennzeichnend für die erste Gruppe ist, daß sie intern strukturiert sind, d.h. auf einer bestimmten Lernstufe einsetzen und über mehr oder weniger viele Zwischenstufen auf ein höheres Sprachniveau führen. Dagegen verfolgen die ergänzenden Lernmittel eng umgrenzte Ziele, die sowohl auf der inhaltlichen als auch auf der methodischen Seite liegen können. Sie dienen der Erweiterung oder Vertiefung bestimmter Aspekte oder schlichtweg zur methodischen Auflockerung. Die Bestimmung obligatorischer Steuerungselemente kann sich daher nur auf tragende Lernprogramme beziehen, denen eine zentrale Stellung im Prozeß des Fremdsprachenerwerbs zugedacht ist.

Multimedial angelegte Lernprogramme sollten auf jeden Fall eine Schaltmöglichkeit aufweisen, über die alle akustischen Vorgaben oder Rückmeldungen steuerbar sind. Selbst wenn dann einzelne Programmteile, so z.B. Hörverständnisübungen, ohne die Toneinspielungen als nicht mehr sinnvoll erscheinen, können sie auf diese Weise in Gemeinschaftsarbeitsräumen wenigstens noch eingeschränkt bearbeitet werden, etwa zur Wiederholung.

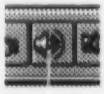

Abb. 6: Tonschalter

Bei fremdsprachenbezogenen Übungsprogrammen ist häufig die Eingabe von Zeichen erforderlich, die auf der in vielen Ländern zur Verfügung stehenden Tastatur nicht vorgesehen sind. Weichen Tastenbeschriftung und benötigte Tastenbelegung in vielen Fällen voneinander ab, dürfte die Einblendung einer sogenannten virtuellen Tastatur auf dem Bildschirm die beste Lösung sein. Beschränken sich die Abweichungen auf einige wenige Zeichen, wie etwa bei Lernprogrammen für das Deutsche auf die Umlaute und das ß, dann reichen einige zusätzliche Bildschirmtasten mit nur geringem Platzbedarf aus.

Abb. 7: Bildschirmtasten zur Eingabe von Sonderzeichen

Hinweise zur Programmbedienung sollten auf jeden Fall in irgendeiner Form auf dem Bildschirm abrufbar sein, um in Zweifelsfällen das Nachschlagen im oft am Arbeitsplatz nicht (mehr) greifbare Handbuch entbehrlich zu machen. Neben ihrer Funktion als Nothelfer geben diese Hinweise auch Aufschluß über die Inhalte und Lernziele des Programms, können also für eher bedächtige Lerntypen einen Ersatz für das Ausprobieren der vorhandenen Programmteile abgeben. Soweit umsetzbar sollten die Hilfen kontextsensitiv angelegt sein, damit Bedienungsprobleme nur zu möglichst geringfügigen Unterbrechungen der Lerntätigkeit führen.

Häufig unentbehrlich ist eine Zugangsmöglichkeit zu Lösungen der jeweils aktuellen Aufgabe, wenngleich dieser Zugang nicht zu jeden Übungszeitpunkt auch offenstehen müßte. Je nach Aufgabengestaltung kann die Lösungsoption erst nach einem oder einigen Versuchen aktivierbar sein. Dieses vorübergehende Sperren widerspricht auch nicht dem Prinzip der Selbstbestimmung beim autonomen Lernen, denn durch diese Maßnahme wird niemand gezwungen, festgelegte Handlungen zum Zwecke des Lernens auszuüben.

Im Grenzbereich zwischen obligatorischen und fakultativen Elementen liegt eine sich ständig aktualisierende Hinweiszeile zu den zum aktuellen Bearbeitungspunkt offenstehenden Optionen. Diese knappen, maximal einzeiligen Anleitungen haben sich in vielen Anwendungen etabliert und werfen schon wegen des geringen Platzbedarfs kein großes Entscheidungsproblem auf. Die Hinweise selbst sind vor allem für Programmneulinge nützlich, weshalb sich ihr Einsatz in Lernprogrammen mit ihrer relativ kurzen Nutzungsdauer besonders empfiehlt. Da sie oft schon nach wenigen Durchgängen überflüssig werden, sollten sie eher dezent gestaltet werden, um einem unruhigen Gesamteindruck der Oberfläche zu vermeiden.

Was die Zuordnung zu einer der beiden Elementgruppen angeht, liegen die Verhältnisse bei einem Feld für Kommentare klarer. In vielen Übungen reicht ein einfacher Richtig-Falsch-Hinweis als Rückmeldung aus, bei schriftlichen Eingaben gegebenenfalls noch ergänzt um eine optische Hervorhebung problematischer Stellen. Gelegentlich sind jedoch auch differenziertere Bewertungen, ausführlichere Hinweise, Erklärungen, Verweise auf Alternativen oder zusätzliche Informationen, Ergebniskommentare oder sonstige Anmerkungen notwendig. Da diese Meldungen trotz unterschiedlicher Einzelfunktionen einem Zweck dienen, nämlich dem der Kommentierung der erbrachten bzw. nicht erbrachten Leistung, sollten sie nach dem Grundsatz, Gleiches gleich zu behandeln, zumindest innerhalb eines Programms einheitlich dargestellt werden.

Ein fest in die Oberfläche eingefügtes Steuerelement dürfte in vielen Fällen nicht die beste Lösung sein. Zum einen schon deshalb, weil nicht zu allen Aufgaben Kommentare nötig oder möglich sind, und zum anderen, weil sich die Art der Kommentierung in kein Schema pressen läßt, was Aussagen zu ihrem Umfang erschwert. Ein in allen Programmteilen reservierter Platz, dessen Ausmaße sich nach dem ausführlichsten Hinweis zu richten hätten, dürfte in der Regel zuviel Fläche beanspruchen. Was sich anbietet ist also ein je nach Bedarf

sichtbares, positioniertes und dimensioniertes Feld, das durch ein gleichbleibendes Erscheinungsbild auf den Charakter seiner Anzeigen verweist.

Den Gleichheitsgrundsatz und die Vorteile der Analogiebildung nutzt ein nach Art der Tastenbeschriftung von Video- und Tonkassettenrekordern gestaltetes Bedienfeld zur Steuerung von Ton- und Bildsequenzen. Sofern nicht durchgängig benötigt, was für einen Platz in der Steuerleiste spräche, stellt eine möglichst nahe Positionierung an dem Element, das es steuern soll, eine geeignete Lösung dar.

Alternativ dazu kann aber auch eine zusätzliche Steuerleiste eingeführt werden, die alle oder die meisten fakultativen Steuerelemente aufnimmt. Die Sammlung dieser Elemente in einem eigenen Container erlaubt es, aus dieser Anordnung Rückschlüsse auf ihre Funktion und damit ihre Einsatzbereiche zu ziehen.

Nicht zuletzt dürfte eine Titelzeile, die entweder den Programmnamen oder denjenigen der gerade aktiven Übung anzeigt, auch in den meisten Lernprogrammen keine Platzverschwendung verursachen.

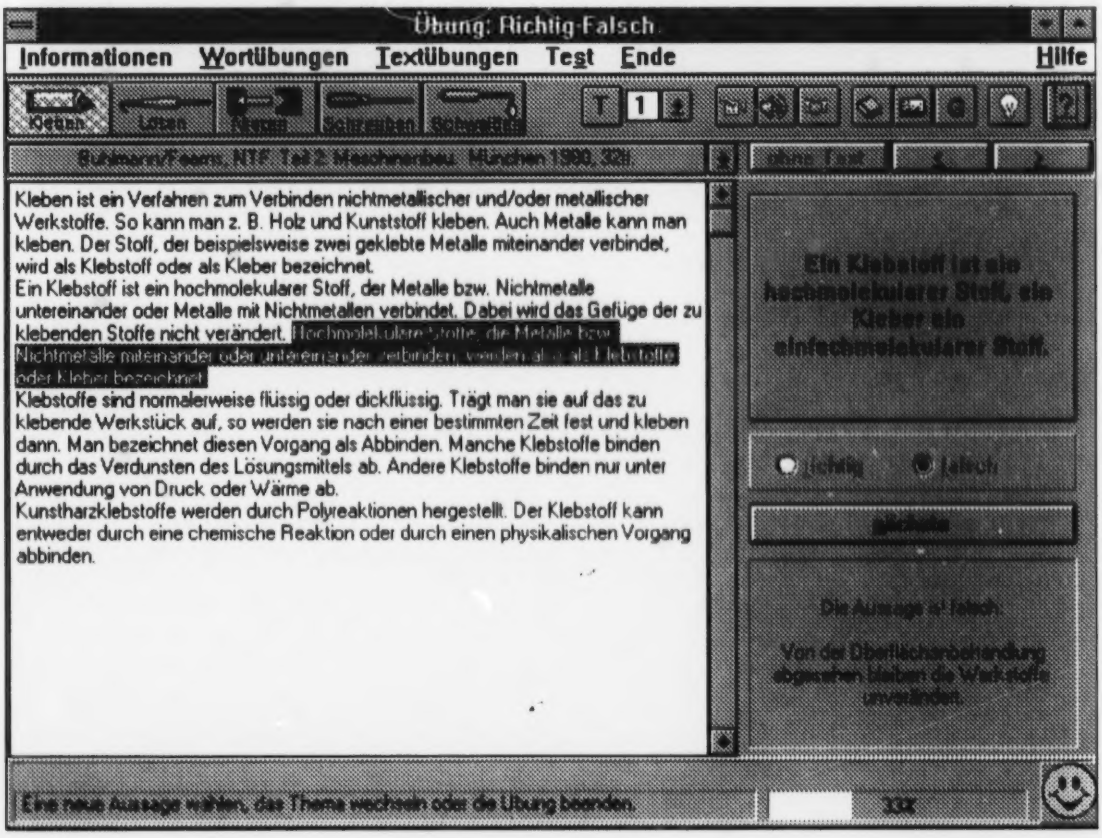
Übung: Richtig-Falsch
Informationen
Wortübungen
Textübungen
Test
Ende
Hilfe
Kleben ist ein Verfahren zum Verbinden nichtmetallischer und/oder metallischer Werkstoffe. So kann man z. B. Holz und Kunststoff kleben. Auch Metalle kann man kleben. Der Stoff, der beispielsweise zwei geklebte Metalle miteinander verbindet, wird als Klebstoff oder als Kleber bezeichnet.
Ein Klebstoff ist ein hochmolekularer Stoff, der Metalle bzw. Nichtmetalle untereinander oder Metalle mit Nichtmetallen verbindet. Dabei wird das Gefüge der zu klebenden Stoffe nicht verändert.
Klebstoffe sind normalerweise flüssig oder dickflüssig. Trägt man sie auf das zu klebende Werkstück auf, so werden sie nach einer bestimmten Zeit fest und kleben dann. Man bezeichnet diesen Vorgang als Abbinden. Manche Klebstoffe binden durch das Verdunsten des Lösungsmittels ab. Andere Klebstoffe binden nur unter Anwendung von Druck oder Wärme ab.
Kunstharzklebstoffe werden durch Polyreaktionen hergestellt. Der Klebstoff kann entweder durch eine chemische Reaktion oder durch einen physikalischen Vorgang abbinden.
Richtig
Falsch

Abb. 9: Oberflächenaufbau des Lernprogramms *Verbindungsarten*, Übung *Richtig-Falsch*

BEST COPY AVAILABLE

5. Perspektive

Die Reihe fakultativer Steuerelemente ließe sich vermutlich noch eine Weile fortführen, wird jedoch immer offen bleiben, schon um weiteren Entwicklungen Raum zu lassen. Und auch bei den obligatorischen könnten Ergänzung notwendig werden. Es sollte hier auch keine definitive Bestimmung oder gar Normierung angestrebt werden. Es geht vielmehr darum, Anstöße zu einer Linienführung zu geben, die letztendlich einen Großteil unnötiger Mehrarbeit vermeiden könnte.

Eine verläßliche Anforderungsbestimmung und Qualitätsbeschreibung von Status- und Steuerelementen für Lernprogramme hätte den Vorteil, daß sich der Entwicklungsaufwand dafür genauer kalkulieren und darüber oft erst realisieren ließe. Außerdem könnte für weite Bereiche verhindert werden, daß sich jeder Entwickler wieder dieselben Fragen stellt, nur um womöglich auch wieder nur dieselben Antworten wie schon einige vor ihm zu finden. Gerade die Entwicklung von sprachbezogenen Lernprogrammen leidet immer noch unter dem Phänomen des Einzelkämpfertums (Räkel/Steinfeld 1992:145), das eine voranschreitende, zeitgemäße Lernprogrammierung behindert. Vielleicht leisten die hier vorgestellten Überlegungen einen Beitrag zur Überwindung der offensichtlich vorhandenen Schwellen.

Literatur:

Brücher, K. H. (1991): Autorenprogramme im computergestützten Fremdsprachenunterricht. In: Deutsch als Fremdsprache 3, 175-180

Mitschian, H. (1993): Objektorientierte Programmierung: Perspektiven für computergestützte Lern- und Lehrprogramme DaF. In: Unterrichtswissenschaft 3, 261-280

Räkel H.-H. S. / Steinfeld T. (1992): Vorwort zur 'Dokumentation zum computergestützten Unterricht in Deutsch als Fremdsprache'. In: Info DaF 2, 145